IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: **DeWitt, Jr. et al.**	§	
	§	Group Art Unit: **2193**
Serial No.: **10/675,776**	§	
	§	Examiner: **Vu, Tuan A.**
Filed: **September 30, 2003**	§	
	§	Confirmation No.: **6262**
For: **Method and Apparatus for Counting Execution of Specific Instructions and Accesses to Specific Data Locations**	§	

35525

PATENT TRADEMARK OFFICE
CUSTOMER NUMBER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF (37 C.F.R. 41.37)

This brief is in furtherance of the Notice of Appeal filed concurrently herewith on September 30, 2009.

A fee of $540.00 is required for filing an Appeal Brief. Please charge this fee to IBM Corporation Deposit Account No. 09-0447. No additional fees are believed to be necessary. If, however, any additional fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447.

REAL PARTY IN INTEREST

The real party in interest in this appeal is the following party: International Business Machines Corporation of Armonk, New York.

RELATED APPEALS AND INTERFERENCES

This appeal has no related proceedings or interferences.

STATUS OF CLAIMS

A. TOTAL NUMBER OF CLAIMS IN APPLICATION

The claims in the application are: 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53

B. STATUS OF ALL THE CLAIMS IN APPLICATION

Claims canceled: 2-5, 7-25, 27, 28, 31, 33, 36, 37, and 40-48

Claims withdrawn from consideration but not canceled:

Claims pending: 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53

Claims allowed: None

Claims rejected: 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53

Claims objected to: None

C. CLAIMS ON APPEAL

The claims on appeal are: 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53

STATUS OF AMENDMENTS

No amendments were filed after the Final Office Action mailed June 30, 2009.

SUMMARY OF CLAIMED SUBJECT MATTER

A. CLAIM 1 - INDEPENDENT

The subject matter of claim 1 is directed to a computer implemented method in an instruction cache of a data processing system for monitoring execution of instructions. The method comprises receiving a bundle at an instruction cache unit (Fig. 3; specification, page 23, paragraph 1), the bundle containing at least one instruction slot, (page 26, paragraph 2, FIG. 5, reference numeral 502, 504, 506) wherein the instruction slot contains an instruction (page 27, paragraph 1); responsive to receiving the bundle, determining by the instruction cache unit whether the bundle contains an indicator within at least one spare bit of the at least one instruction slot (page 27, paragraph 1), wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit (page 29, paragraph 1); responsive to a determination that the bundle contains the indicator within the at least one instruction slot, sending a signal by the instruction cache unit to a performance monitor unit (page 29, paragraph 1), wherein upon receiving the signal, the performance monitor unit increments a counter in the instruction cache unit that is associated with the instruction (page 31, paragraph 4), the incrementing providing a count of a number of times the instruction is executed (page 23, paragraph 2; page 43, paragraph 2; page 46, paragraph 1); and sending the bundle from the instruction cache unit to a functional unit for execution of the instruction (page 31, paragraph 4).

B. CLAIM 32 - INDEPENDENT

The subject matter of claim 32 is directed to a computer program product that comprises a computer recordable medium having computer useable program code for monitoring execution of instructions (page 64, paragraph 1), the computer program product comprises computer usable program code (page 64, paragraph 1) for receiving a bundle at an instruction cache unit (Fig. 3; specification, page 23, paragraph 1), the bundle containing at least one instruction slot (page 26, paragraph 2, FIG. 5, reference numeral 502, 504, 506), wherein the instruction slot contains an instruction (page 27, paragraph 1); computer usable program code (page 64, paragraph 1) for, responsive to receiving the bundle, determining by the instruction cache unit whether the bundle contains an indicator within at least one spare bit of the at least one instruction slot (page 27,

paragraph 1), wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit (page 29, paragraph 1); computer usable program code (page 64, paragraph 1) for, responsive to a determination that the bundle contains the indicator within the at least one instruction slot, sending a signal by the instruction cache unit to a performance monitor unit (page 29, paragraph 1), wherein upon receiving the signal, the performance monitor unit increments a counter in the instruction cache unit that is associated with the instruction (page 31, paragraph 4), wherein the incrementing provides a count of a number of times the instruction is executed (page 23, paragraph 2; page 46, paragraph 1); and computer usable program code (page 64, paragraph 1) for, sending the bundle from the instruction cache unit to a functional unit for execution of the instruction (page 31, paragraph 4).

C. CLAIM 49 - INDEPENDENT

The subject matter of claim 49 is directed to a computer-implemented method of monitoring software performance in a data processing system (page 20, paragraph 2). The computer-implemented method comprises detecting an indicator associated with one of an instruction and a memory location unit of a processor (page 32, paragraph 2); responsive to detecting the indicator, incrementing a counter in an instruction cache unit that is associated with the indicator (page 45, paragraph 1); and analyzing, in a performance monitor unit, a value of the counter to determine a performance of the data processing system (page 33, paragraph 4).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The grounds of rejection to review on appeal:

A. GROUND OF REJECTION 1

Whether the Office Action fails to establish that claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53 lack a well established utility which would cause one skilled in the art to be unable to make and use the claimed invention as required under 35 U.S.C. § 112, first paragraph.

B. GROUND OF REJECTION 2

Whether the Office Action fails to establish a *prima facie* case of obviousness under 35 U.S.C. § 103 against claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53 over *Gover et al.*, ("*Gover*"), U.S. Patent 5,752,062, and further in view of APA (Admitted Prior Art: Specifications Background: pg. 3-4).

ARGUMENT

A. SUMMARY OF ARGUMENT

The statutory rejections under 35 U.S.C. § 112, first paragraph, and 35 U.S.C. § 103 are substantively and legally erroneous and must be reversed. These statutory rejections require at least a *prima facie* showing based on a preponderance of evidence to support a conclusion that the claims lack utility and are obvious over the cited art of *Gover*. Here, in this case, the evidentiary support which will be considered in detail, *infra*, is so sketchy and insufficient that credible rejections based on the code sections referenced are simply erroneous.

The Office Action rejects the claims under 35 U.S.C. § 112, first paragraph, on grounds of insufficient utility. Procedurally, the rejection under this code section is improper. Rejections asserting lack of substantial utility are grounded in 35 U.S.C. § 101 may, at times, include sufficient evidence to provide a basis for an accompanying 35 U.S.C. § 112, first paragraph, rejection. However, in the final Office Action of June 30, 2009, no claims are rejected under 35 U.S.C. § 101, nor does any evidence exist to support a credible assertion of a utility rejection under 35 U.S.C. § 101. Nevertheless, the final Office Action includes rejection under 35 U.S.C. § 112, first paragraph, ***solo***, on grounds of insubstantial utility and asserts in a lengthy diatribe that the disclosure lacks specific functional details and description for the recited claim feature "instruction cache unit." An instruction cache unit is a well-known structure to one skilled in the art and it is sufficiently disclosed in the specification to satisfy the enablement requirement under 35 U.S.C. § 112, first paragraph. Therefore, the rejection under 35 U.S.C. § 112, first paragraph, is substantively erroneous, in addition to being procedurally erroneous.

Further, the Office Action relies on the cited art of *Gover* in view of Admitted Prior Art to reject the pending claims of the application. The fundamental distinction between *Gover* and the application is the existence of an ***instruction cache unit*** that is required by the claim language. *Gover* discloses an ***instruction cache*** that has no processing capability. By contrast, the claims require an ***instruction cache unit*** that incorporates processing capabilities. This fundamental distinction makes it virtually impossible to posit that *Gover* is even remotely similar to the claimed features of the subject disclosure.

Despite the interviews held on this matter involving the inventor since April, 2009 and Office Action responses subsequent thereto that try to further clarify the meaning and function of an instruction cache unit, prosecution remains at a stalemate with Office Actions subsequent to April, 2009 continuing to assert confusion and a lack of understanding of the meaning of instruction cache unit, *inter alia*. The assertions of the Office Action are not reasonable and are further contrary to statute, case law, rule, and Patent Office procedure. Accordingly, based on the arguments presented, *infra*, Appellant respectfully requests that the rejections be reversed, *in toto*, and the claims be allowed to issue.

B. GROUND OF REJECTION 1 (Claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53)

The first ground of rejection asserts that claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53 lack a well established utility which would cause one skilled in the art to be unable to use the claimed invention as required under 35 U.S.C. § 112, first paragraph. This rejection is erroneous and requires reversal.

The Office Action rejects the claims under this code section and states in pertinent part:

> Claims, 1, 6, 26, 29-30, 32, 34-35, and 38-39 are rejections for NOT providing a deliberate, fundamental and clear implementation detail regarding the "instruction cache unit" in terms of utility or capacity to cache instruction OR structural details corroborating that this "instruction cache unit" is equipped with processing capability.
>
> **The 'instruction cache unit' as disclosed is not equipped with cache capability nor is it equipped with processor functionality from the entire disclosure.** There is no explicit description as to how instruction has been cached in the "instruction cache unit", i.e. an established entity resident to the disclosed 'instruction cache unit' that actually maintain any cached data, as the nomenclature entails. Nor are there sufficient processor capabilities (from the entire Disclosure) established inside the 'instruction cache unit' to effectively corroborate to any claimed functionality of 'instruction cache unit' as recited or even as illustrated in the Drawings (see Fig. 31 and text).
>
> According to the analysis of the Specifications the "instruction cache unit" is nowhere taught as explicitly destined to cache instructions but rather to operate as a processing entity reminiscent of one equipped with programmatic or hardware functionality in terms of executed actions of *processing, determining* then *sending*. In this light however, there no sufficient details reminiscent of a processor element or software entity established inside this 'instruction cache unit' in order for the

'instruction cache unit' 3106 (see Fig. 31 – registers 3108, counter 3110) to *determine, process and send* as claimed.

Final Office Action of June 30, 2009, pages 11-12, paragraph 11 (emphasis in original).

Instruction cache units are known to those skilled in the art. For example, one of ordinary skill would acknowledge the existence of at least an IBM® Power2 Instruction Cache Unit and instruction cache units within an IBM® Power PC 405 processor. The disclosure of Appellant provides an instruction cache unit that comprises the primary functions of processing, determining, and sending information. The disclosure provides in pertinent part:

> Turning now to **Figure 3**, a diagram illustrating components used in processing instructions associated with indicators is depicted in accordance with a preferred embodiment of the present invention. Instruction cache unit **300** receives bundles **302**. Instruction cache unit **300** is an example of instruction cache unit **214** in **Figure 2**. A bundle is a grouping of instructions. This type of grouping of instructions is typically found in an IA-64 processor, which is available from Intel Corporation. Instruction cache unit **300** processes instructions for execution.

Specification, page 22, paragraph 2.

The disclosure further provides:

> Turning now to **Figure 7**, a flowchart of a process for processing instructions containing performance indicators is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 7** may be implemented in an instruction cache unit, such as instruction cache unit **214** in **Figure 2**.

Specification, page 28, paragraph 2.

Still further, the Specification states in pertinent part:

> Turning now to **Figure 3**, a diagram illustrating components used in processing instructions associated with indicators is depicted in accordance with a preferred embodiment of the present invention. Instruction cache unit **300** receives bundles **302**. Instruction cache unit **300** is an example of instruction cache unit **214** in **Figure 2**. A bundle is a grouping of instructions. This type of grouping of instructions is typically found in an IA-64 processor, which is available from Intel Corporation. Instruction cache unit **300** processes instructions for execution.

Specification, page 22, paragraph 2.

Additionally, the disclosure provides:

> Turning now to **Figure 7**, a flowchart of a process for processing instructions containing performance indicators is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 7** may be implemented in an instruction cache unit, such as instruction cache unit **214** in **Figure 2**.

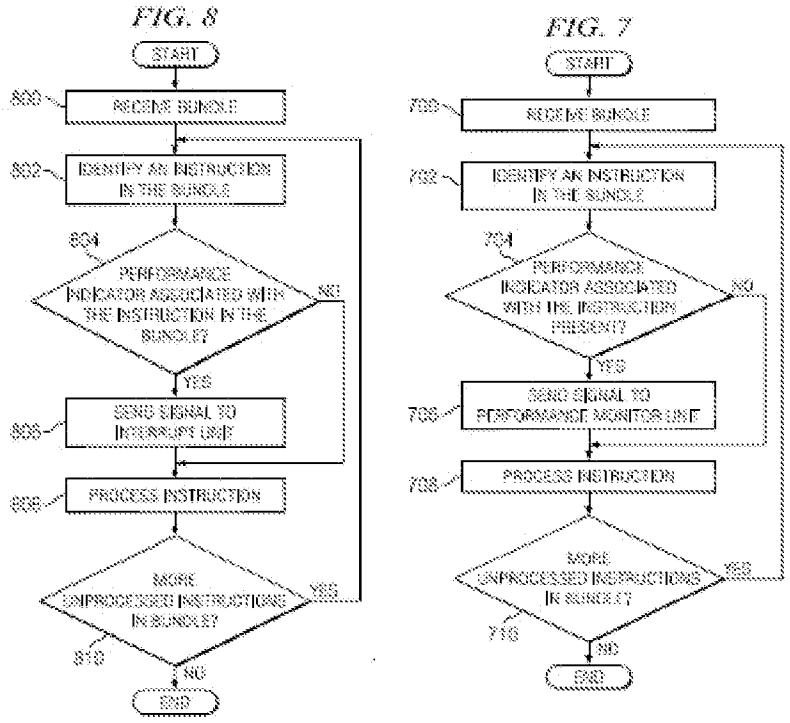
Specification, page 28, paragraph 2.

Still further, the disclosure states in pertinent part:

> Turning now to **Figure 8** a flowchart of a process for selectively sending signals to an interrupt unit is depicted in accordance with a preferred embodiment of the present invention. The process illustrated in **Figure 8** may be implemented in an instruction cache unit, such as instruction cache unit **242** in **Figure 2**. This process is employed in cases in which monitoring events using a performance monitor unit may miss certain events. For example, a performance monitor unit counts events. When a cache miss occurs, a signal is sent to the performance monitor unit. When the meta data for a corresponding cache line is loaded into the cache, the appropriate signal or signals also are raised. If the meta data indicates that an exception is to be raised, then a signal is sent to the interrupt unit in which the signal indicates that an exception is to be raised.

Specification, page 29, paragraph 3.

Additionally, at least **Figures 7** and **8**, referenced in the citations, *supra*, disclose processes that may be implemented by an instruction cache unit. **Figures 7** and **8** are reproduced below, *infra*, for reference. Therefore, the disclosure of Appellant enables the feature instruction cache unit.



It is well known that a patent application satisfies the enablement requirement if "one skilled in the art, after reading their disclosures, could practice the invention claimed … without undue experimentation." *In re* Wands, 858 F.2d 731, 735, 8 USPQ2d 1400 (Fed. Cir. 1988). "However, this is not to say that the specification itself must necessarily describe how to make and use every possible variant of the claimed invention, for the artisan's knowledge of the prior art and routine experimentation can often fill in gaps, interpolate between embodiments and perhaps even extrapolate between the disclosed embodiments, depending upon the predictability of the art." *AK Steel Corp. v. Sollac*, 344 F.3d 1234, 68 USPQ2d 1280 (Fed. Cir. 2003). A patent disclosure need not enable information within the knowledge of an ordinarily skilled artisan. Thus a patentee preferably omits from the disclosure any routine disclosure that is well known at the time of the application. *Chiron Corporation v. Genentech, Inc.* 363 F.3d 1247, 70 USPQ2d 1321 (Fed. Cir. 2004). In this case, the featured instruction cache unit may be implemented by one of ordinary skill in the art without undue experimentation. The disclosure provides that the instruction cache unit includes additional functional blocks such as a counter and range registers in order to perform the

process considered novel to the disclosure. **Figure 31** of the disclosure further illustrates an instruction cache unit that implements features that include range registers and a counter.

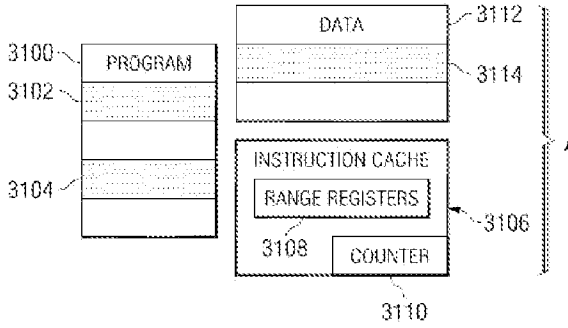


FIG. 31

No additional disclosure is required to support the functionality of the instruction cache unit as recited in the claims and detailed in the disclosure. Accordingly, based on the foregoing, the rejection of the claims under 35 U.S.C. § 112, first paragraph, constitutes legal error and it is respectfully requested that this rejection be reversed.

C. GROUND OF REJECTION 2 (Claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53)

In another ground of rejection, claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53 are rejected as obvious over *Gover* in view of Admitted Prior Art. However, the rejections are not based on a conclusion of *prima facie* obviousness. Therefore, the rejection is erroneous and requires reversal.

Claim 1 is representative and reproduced below:

> 1. A computer implemented method in an instruction cache of a data processing system for monitoring execution of instructions, the method comprising:
>
> receiving a bundle at an instruction cache unit, the bundle containing at least one instruction slot, wherein the instruction slot contains an instruction;
>
> responsive to receiving the bundle, determining by the instruction cache unit whether the bundle contains an indicator within at least one spare bit of the at least one instruction slot, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;
>
> responsive to a determination that the bundle contains the indicator within the at least one instruction slot, sending a signal by the instruction cache unit to a performance monitor unit, wherein upon receiving the signal, the performance monitor unit increments a counter in the instruction cache

unit that is associated with the instruction, the incrementing providing a count of a number of times the instruction is executed; and

sending the bundle from the instruction cache unit to a functional unit for execution of the instruction.

At a minimum, claim 1 requires the existence of an ***instruction cache unit***. However, the combination of *Gover* and the admitted prior art fail to teach or fairly suggest an ***instruction cache unit*** as recited in claim 1 and enabled by the disclosure. At best, *Gover* only discloses an instruction cache that stores instructions in accordance with the functionality of instruction caches well known to one skilled in the art.

Specifically, the instruction cache of *Gover* stores instructions. *Gover* discloses in pertinent part:

> If one or more of the sequence of instructions is not stored in instruction cache 14, then **instruction cache** 14 inputs (through BIU 12 and system bus 11) such instructions from system memory 39 connected to system bus 11. In response to the instructions input from **instruction cache** 14, sequencer unit 18 selectively dispatches through a dispatch unit 46 the instructions to selected ones of execution units....

Gover, column 5, lines 24-34 (emphasis added).

Additional traditional uses of the instruction cache are disclosed throughout *Gover*. For example, *Gover* further discloses:

> In the fetch stage. sequencer unit 18 **selectively inputs (from instructions cache 14) one or more instructions** from one or more memory addresses storing the sequence of instructions discussed further hereinabove in connection with branch unit 20 and sequencer unit 18. The decode stage, sequencer unit 18 decodes up to four fetched instructions.

Gover, column 6, lines 7-12 (emphasis added).

By contrast, the claims feature an ***instruction cache unit*** that includes additional logic for processing data, *e.g.* counting, selecting, and transmitting data. One representation of the disclosed instruction cache unit recited in the claim is illustrated in Figure 31, reproduced *infra*.

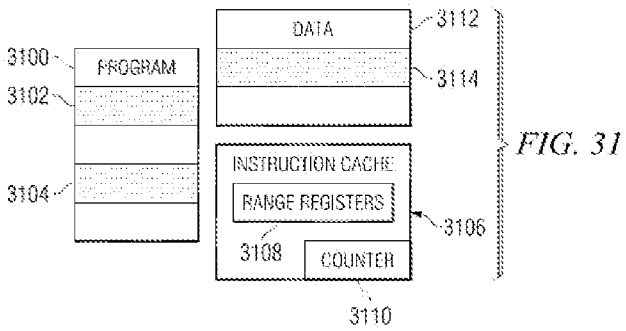


FIG. 31

In FIG. 31, instance 3106 of the disclosure illustrates an instruction cache unit that includes a counter 3110, and range registers 3108 that may function in conjunction with counter 3110. Accordingly, it must be noted that the instruction cache unit possessed components that enable a processing capability which exceeds the functionality of the common **instruction cache** disclosed in *Glover*. For at least this reason along, the cited art of Glover cannot read on the claims. The asserted Admitted prior art, *i.e.* pages 2-3 of the specification, adds nothing to cure the deficiencies in *Glover*. Nowhere in the background of the disclosure of *Glover* is there any reference to an **instruction cache unit** that performs any type of processing as recited in the claims.

In summary, *Glover* and the asserted admitted prior art do not collectively or individually teach or fairly suggest an ***instruction cache unit***. Therefore, for at least this reason a conclusion of *prima facie* obviousness cannot be established. It is well-settled that to establish a prima facie case of obviousness, all the claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981,985, 180 USPQ 580 (CCPA 1974).

"Rejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.'" *KSR Int'l*, 550 U.S. 398, 82 USPQ2d 1385 (2007) quoting *In re Kahn*, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336 (Fed. Cir. 2006). Based on the foregoing in this case, there is no evidence that would support a conclusion of *prima facie* obviousness. Accordingly, the rejection under 35 U.S.C. § 103 is erroneous and it is respectfully requested that this rejection be reversed.

D. CONCLUSION

The rejection of claims 1, 6, 26, 29, 30, 32, 34, 35, 38, 39, and 49-53 under 35 U.S.C. § 112, first paragraph, and under 35 U.S.C. § 103(a) by *Gover* and the Admitted Prior Art is substantively and illegally erroneous. Therefore, Appellant respectfully requests that the Board of Patent Appeals and Interferences reverse the rejections so that an allowance of this application may be entered.

Date: September 30, 2009

Respectfully submitted,

/A. M. Thompson/

A. M. Thompson
Reg. No. 59,673
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777

CLAIMS APPENDIX

The text of the claims involved in the appeal is as follows:

1. A computer implemented method in an instruction cache of a data processing system for monitoring execution of instructions, the method comprising:

receiving a bundle at an instruction cache unit, the bundle containing at least one instruction slot, wherein the instruction slot contains an instruction;

responsive to receiving the bundle, determining by the instruction cache unit whether the bundle contains an indicator within at least one spare bit of the at least one instruction slot, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;

responsive to a determination that the bundle contains the indicator within the at least one instruction slot, sending a signal by the instruction cache unit to a performance monitor unit, wherein upon receiving the signal, the performance monitor unit increments a counter in the instruction cache unit that is associated with the instruction, the incrementing providing a count of a number of times the instruction is executed; and

sending the bundle from the instruction cache unit to a functional unit for execution of the instruction.

6. The computer implemented method of claim 1, wherein the counter is located in a shadow memory.

26. The computer implemented method of claim 1, wherein the bundle contains the indicator in a spare field of the bundle.

29. The computer implemented method of claim 1, further comprising:

responsive to a determination that the bundle contains the indicator, incrementing the counter, wherein the counter tracks any subsequent instruction executed by an associated processor.

30. The computer implemented method of claim 29, wherein the bundle is a first bundle, the method further comprising:

receiving a second bundle at the instruction cache;

responsive to receiving the second bundle, determining whether a second instruction in the second bundle contains a second indicator; and

responsive to a determination that the second bundle contains the second indicator, ending the incrementing of the counter.

32. A computer program product comprising:

a computer recordable medium having computer useable program code for monitoring execution of instructions, the computer program product comprising:

computer usable program code for receiving a bundle at an instruction cache unit, the bundle containing at least one instruction slot, wherein the instruction slot contains an instruction;

computer usable program code for, responsive to receiving the bundle, determining by the instruction cache unit whether the bundle contains an indicator within at least one spare bit of the at least one instruction slot, wherein the indicator identifies the instruction as one that is to be monitored by a performance monitor unit;

computer usable program code for, responsive to a determination that the bundle contains the indicator within the at least one instruction slot, sending a signal by the instruction cache unit to a performance monitor unit, wherein upon receiving the signal, the performance monitor unit increments a counter in the instruction cache unit that is associated with the instruction, wherein the incrementing provides a count of a number of times the instruction is executed; and

computer usable program code for, sending the bundle from the instruction cache unit to a functional unit for execution of the instruction.

34. The computer program product of claim 32, wherein the computer usable program code for incrementing the counter further comprises computer usable program code for incrementing the counter, wherein the counter is located in a shadow memory.

35. The computer program product of claim 32 further comprising wherein the bundle contains the indicator in a spare field of the bundle.

38. The computer program product of claim 32, further comprising:

computer usable program code, responsive to a determination that the bundle contains the indicator, incrementing the counter, wherein the counter tracks any subsequent instruction executed by an associated processor.

39. The computer program product of claim 38 wherein the bundle is a first bundle, the computer program product further comprising:

computer usable program code for receiving a second bundle at the instruction cache unit;

computer usable program code, responsive to receiving the second bundle, for determining whether a second instruction in the second bundle contains a second indicator; and

computer usable program code, responsive to a determination that the second bundle contains the second indicator, for ending incrementing the counter.

49. A computer-implemented method of monitoring software performance in a data processing system, the computer-implemented method comprising:

detecting an indicator associated with one of an instruction and a memory location unit of a processor;

responsive to detecting the indicator, incrementing a counter in an instruction cache unit that is associated with the indicator; and

analyzing, in a performance monitor unit, a value of the counter to determine a performance of the data processing system.

50. The computer-implemented method of claim 49, wherein responsive to the indicator being associated with an instruction, the incrementing occurs each time an instruction is executed.

51. The computer-implemented method of claim 49, wherein responsive to the indicator being associated with a memory location unit, the incrementing occurs each time the memory location unit is accessed.

52. The computer-implemented method of claim 49, further comprising:

generating an interrupt from an interrupt unit responsive to the value of the counter exceeding a threshold value.

53. The computer-implemented method of claim 49, further comprising:

including a criteria for the counter; and

generating an interrupt from an interrupt unit responsive to meeting the criteria.

EVIDENCE APPENDIX

This appeal brief presents no additional evidence.

RELATED PROCEEDINGS APPENDIX

This appeal has no related proceedings.